

LAPORAN PRAKTIKUM STRUKTUR DATA  
ALGORITHM SORTING PADA JAVA



DOSEN PENGAMPU :

Dr. Wahyudi, M.T.

OLEH :

RIFKI YULIANDRA

NIM.2311532011

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
PADANG

2024

## *Algorithm Sorting Pada Java*

### I. TUJUAN

- 1.1. Memahami apa yang dimaksud dengan Algoritma Sorting.
- 1.2. Memahami cara membuat program sorting menggunakan metode InsertionSort.
- 1.3. Memahami cara membuat program sorting menggunakan metode SelectionSort.
- 1.4. Memahami cara membuat program sorting menggunakan metode BubbleSort.

### II. TEORI

Dalam dunia pemrograman, mengelola data secara efektif merupakan hal yang krusial. Salah satu teknik penting untuk mencapai hal ini adalah dengan menggunakan algoritma pengurutan (sorting). Algoritma pengurutan memungkinkan kita untuk menyusun data berdasarkan kriteria tertentu, seperti urutan numerik, alfabet, atau tanggal. Hal ini sangat bermanfaat untuk berbagai aplikasi, seperti analisis data, pencarian informasi, dan visualisasi data.

Java, sebagai salah satu bahasa pemrograman yang populer, menyediakan berbagai macam algoritma pengurutan yang dapat digunakan untuk menyelesaikan berbagai permasalahan. Masing-masing algoritma memiliki karakteristik dan performanya sendiri, sehingga pemilihan algoritma yang tepat sangat penting untuk mengoptimalkan performa program.

Mempelajari teori di balik algoritma pengurutan di Java bukan hanya penting bagi programmer profesional, tetapi juga bagi para pemula yang ingin memahami cara kerja program secara mendalam. Dengan memahami teori ini, programmer dapat:

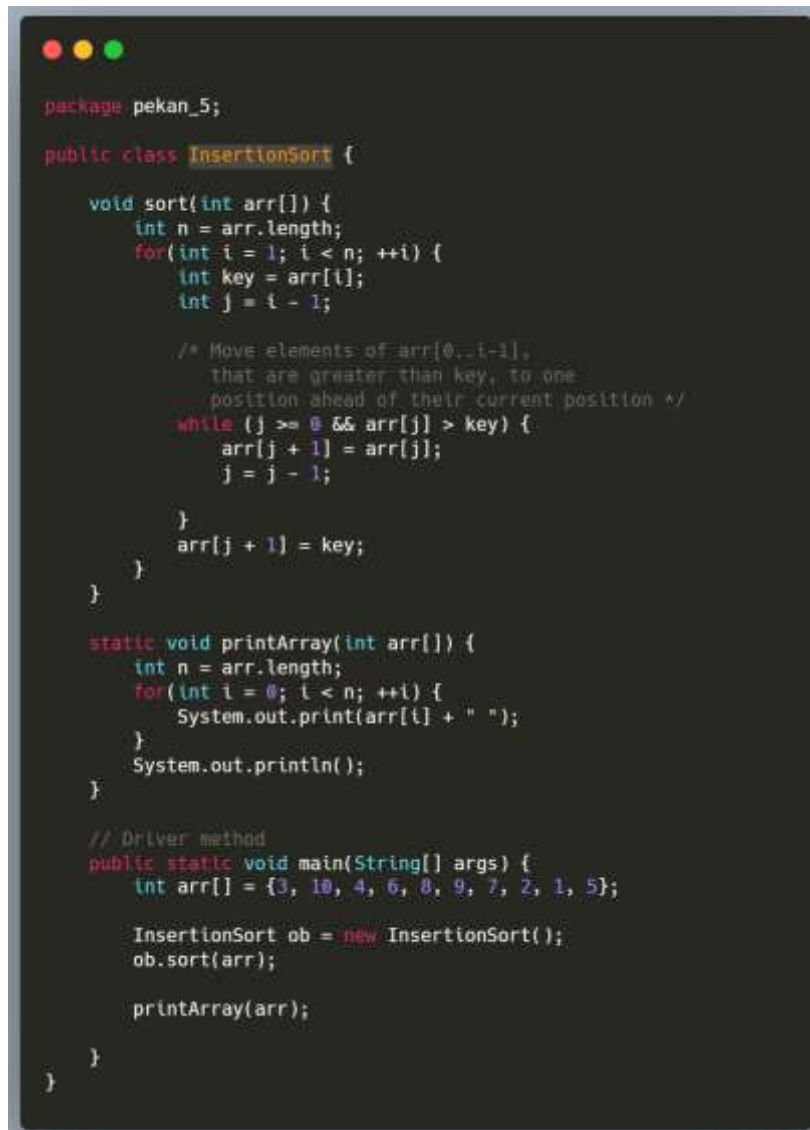
- a. Memilih algoritma yang tepat: Mengetahui kelebihan dan kekurangan masing-masing algoritma memungkinkan programmer untuk memilih algoritma yang paling efisien untuk kebutuhan program mereka.
- b. Menganalisis performa program: Memahami bagaimana algoritma pengurutan bekerja membantu programmer untuk menganalisis performa program dan mengidentifikasi potensi bottleneck.
- c. Mengembangkan algoritma baru: Pengetahuan tentang teori sorting dapat menjadi dasar bagi programmer untuk mengembangkan algoritma pengurutan baru yang lebih optimal untuk kebutuhan spesifik mereka.

Algoritma sorting merupakan salah satu konsep fundamental dalam ilmu komputer dan pemrograman Java. Memahami teori di balik algoritma pengurutan sangat penting

untuk memilih algoritma yang tepat, menganalisis performa program, dan bahkan mengembangkan algoritma baru. Dengan mempelajari teori ini, programmer dapat meningkatkan kemampuan mereka dalam menyelesaikan permasalahan secara efektif dan efisien.

### III. LANGKAH KERJA PRAKTIKUM

#### 3.1. Membuat program InsertionSort



```
package pekan_5;

public class InsertionSort {

    void sort(int arr[]) {
        int n = arr.length;
        for(int i = 1; i < n; ++i) {
            int key = arr[i];
            int j = i - 1;

            /* Move elements of arr[0..i-1],
             that are greater than key, to one
             position ahead of their current position */
            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
    }

    static void printArray(int arr[]) {
        int n = arr.length;
        for(int i = 0; i < n; ++i) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }

    // Driver method
    public static void main(String[] args) {
        int arr[] = {3, 10, 4, 6, 0, 9, 7, 2, 1, 5};

        InsertionSort ob = new InsertionSort();
        ob.sort(arr);

        printArray(arr);
    }
}
```

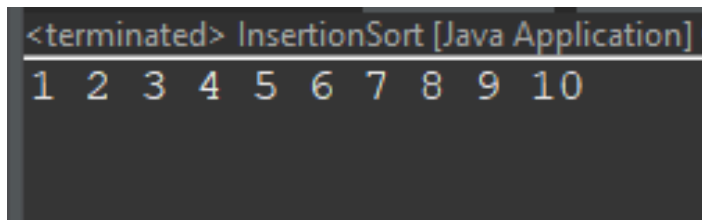
Pada kelas InsertionSort ini berisi dua metode utama yaitu 'sort()' dan 'printArray()'. Metode 'sort()' digunakan untuk melakukan sorting menggunakan algoritma Insertion Sort, sedangkan metode 'printArray()' digunakan untuk mencetak isi array setelah diurutkan.

Metode 'sort()' ini menerima sebuah array integer sebagai argumen dan mengurutkan elemen-elemen dalam array tersebut menggunakan algoritma

Insertion Sort. Algoritma Insertion Sort bekerja dengan cara memilih elemen pertama dalam array sebagai elemen yang akan diurutkan, kemudian memilih elemen berikutnya dan menempatkannya pada posisi yang tepat sehingga elemen-elemen sebelumnya tetap terurut. Proses ini dilakukan terus menerus hingga semua elemen dalam array terurut.

Bagian 'main()' merupakan bagian utama dari program yang dijalankan pertama kali. Dalam bagian ini, dibuat sebuah array integer dengan isi {3, 10, 4, 6, 8, 9, 7, 2, 1, 5}. Kemudian, objek dari kelas 'InsertionSort' dibuat dan metode 'sort()' dipanggil untuk mengurutkan elemen dalam array tersebut. Setelah itu, metode 'printArray()' dipanggil untuk mencetak isi array yang telah diurutkan.

Dan output yang dihasilkan adalah:



```
<terminated> InsertionSort [Java Application] 0
1 2 3 4 5 6 7 8 9 10
```

### 3.2. Contoh lain InsertionSort

```
package pekan_5;

public class InsertionSort2 {

    public static void insertionSort(int[] arr) {
        int n = arr.length;
        for(int i = 1; i < n; i++) {
            int key = arr[i];
            int j = i - 1;
            while(j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j--;
            }
            arr[j + 1] = key;
        }
    }

    public static void main(String[] args) {
        int arr[] = {23, 78, 45, 8, 32, 56, 1};
        int n = arr.length;
        System.out.printf("array yang belum terurut:\n");
        for(int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
        insertionSort(arr);
        System.out.printf("array yang terurut:\n");
        for(int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
}
```

Kelas 'InsertionSort2' ini berisi dua metode utama yaitu insertionSort() dan main(). Metode insertionSort() digunakan untuk melakukan sorting menggunakan algoritma Insertion Sort, sedangkan metode main() digunakan untuk menjalankan program dan mencetak hasil sorting.

Pada kelas ini juga metode insertion sama dengan program 'InsertionSort' yang pertama tadi. Yang mana algoritma ini akan memilih elemen dalam pertama dalam array sebagai elemen yang akan diurutkan, kemudian akan membandingkan dengan elemen berikutnya dan menempatkan sesuai pada ukuran dan posisi yang tepat sehingga elemen-elemen tersebut tetap terurut. Proses ini dilakukan terus-menerus hingga semua elemen dalam array tersebut terurut.

Adapun output yang dihasilkan dari program tersebut:

```
<terminated> InsertionSort2 [Java Application] C
array yang belum terurut:
23 78 45 8 32 56 1
array yang terurut:
1 8 23 32 45 56 78
```

### 3.3. Program dengan metode SelectionSort

```
package pekan_5;

public class SelectionSort {
    public static void selectionSort(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n; i++) {
            int minIndex = i;
            for (int j = i + 1; j < n; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }
            int temp = arr[i];
            arr[i] = arr[minIndex];
            arr[minIndex] = temp;
        }
    }

    public static void main(String[] args) {
        int arr[] = { 23, 78, 45, 8, 32, 56, 1 };
        int n = arr.length;
        System.out.printf("array yang belum terurut:\n");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
        selectionSort(arr);
        System.out.printf("array yang terurut:\n");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
    }
}
```

Kelas 'SelectionSort' ini berisi dua metode utama yaitu selectionSort() dan main(). Metode selectionSort() digunakan untuk melakukan sorting menggunakan algoritma Selection Sort, sedangkan metode main() digunakan untuk menjalankan program dan mencetak hasil sorting.

Metode SelectionSort ini bekerja dengan cara mencari elemen terkecil dalam array dan menukarnya dengan elemen pertama, kemudian mencari elemen kedua terkecil dan menukarnya dengan elemen kedua. Proses ini dilakukan terus menerus hingga semua elemen dalam array tersebut terurut.

Adapun output yang dihasilkan oleh program tersebut adalah:

```
<terminated> SelectionSort [Java Application] C:\P  
array yang belum terurut:  
23 78 45 8 32 56 1  
array yang terurut:  
1 8 23 32 45 56 78
```

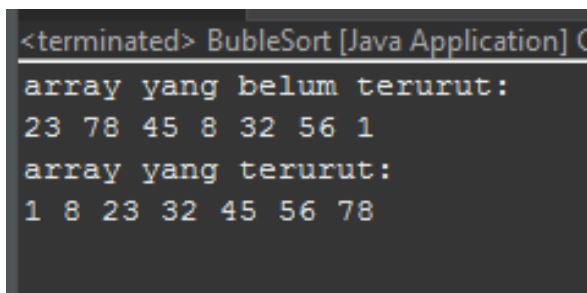
#### 3.4. Program dengan menggunakan metode BubbleSort

```
package pekan_5;  
  
public class BubleSort {  
    public static void bubbleSort(int[] arr) {  
        int n = arr.length;  
        for (int i = 0; i < n; i++) {  
            for (int j = 0; j < n - 1; j++) {  
                if (arr[j] > arr[j + 1]) {  
                    int temp = arr[j];  
                    arr[j] = arr[j + 1];  
                    arr[j + 1] = temp;  
                }  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        int arr[] = { 23, 78, 45, 8, 32, 56, 1 };  
        int n = arr.length;  
        System.out.printf("array yang belum terurut:\n");  
        for (int i = 0; i < n; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println("");  
        // minMaxSelectionSort(arr, n);  
        bubbleSort(arr);  
        System.out.printf("array yang terurut:\n");  
        for (int i = 0; i < n; i++) {  
            System.out.print(arr[i] + " ");  
        }  
        System.out.println("");  
    }  
}
```

Kelas 'BubleSort' ini berisi dua metode utama yaitu bubbleSort() dan main(). Metode bubbleSort() digunakan untuk melakukan sorting menggunakan algoritma Bubble Sort, sedangkan metode main() digunakan untuk menjalankan program dan mencetak hasil sorting.

Metode ini menerima array integer sebagai argumen dan mengurutkan elemen-elemen dalam array tersebut dengan cara membandingkan elemen berurutan dan menggantinya jika urutannya salah. Proses ini dilakukan terus menerus hingga semua elemen dalam array tersebut terurut.

Adapun output yang dihasilkan dari program tersebut adalah:



```
<terminated> BubleSort [Java Application] C
array yang belum terurut:
23 78 45 8 32 56 1
array yang terurut:
1 8 23 32 45 56 78
```

#### IV. KESIMPULAN

Program-program yang telah diberikan adalah implementasi dari tiga algoritma sorting yang berbeda, yaitu Selection Sort, Insertion Sort, dan Bubble Sort. Ketiga algoritma ini digunakan untuk mengurutkan elemen-elemen dalam sebuah array integer. Program-program ini menunjukkan bahwa dengan menggunakan algoritma sorting, kita dapat mengurutkan elemen-elemen dalam array dengan benar dan efisien.

Dalam program Selection Sort, algoritma ini bekerja dengan cara mencari elemen terkecil dalam array dan menukarnya dengan elemen pertama, kemudian mencari elemen kedua terkecil dan menukarnya dengan elemen kedua, dan seterusnya. Sedangkan dalam program Insertion Sort, algoritma ini bekerja dengan cara membandingkan elemen berurutan dan menukarnya jika urutannya salah. Algoritma ini mirip dengan cara kita mengurutkan kartu dalam permainan, yaitu dengan membandingkan kartu berurutan dan menukarnya jika urutannya salah. Dan dalam program Bubble Sort, algoritma ini bekerja dengan cara membandingkan elemen berurutan dan menukarnya jika urutannya salah. Kedua algoritma ini memiliki kelebihan dan kekurangan masing-masing, namun keduanya dapat digunakan untuk mengurutkan elemen-elemen dalam array dengan benar.



Dengan demikian, program-program ini dapat membantu kita memahami bagaimana algoritma sorting bekerja dan bagaimana kita dapat menggunakannya untuk mengurutkan elemen-elemen dalam array. Selain itu, program-program ini juga dapat membantu kita memahami konsep dasar dari pemrograman, seperti penggunaan array, perulangan, dan kondisi.